Statistical Natual Language Processing

Bootstrapping

Instructions

- Submit your work home by email (turmo@cs.upc.edu) or hand it directly in paper (Omega building – office: S321, thirth foor)
- Deadline for the homework: December 2nd 2016, 12:00
- For each day of delay, a point will be discounted from the total grade of the exercise.

ACTIVE LEARNING EXERCISE

(Linux is mandatory to make this exercise. You can use Cygwin environment with Windows)

The goal of the exercise it to know whether it is possible to learn a ME model for the corpus f50 with fewer annotated examples but with the same accuracy than the ME model learned by using the whole training data, which will be called ***C*** up to now.

1. **Learn model *C***. Use the encoded corpus f50/train, which is the whole set of annotated examples, to learn a ME model using the megam_i686.opt (megam_opt for Cygwin) executable:

   ./megam_i686.opt -quiet -fvals multiclass train > f50.mem (for linux)
   ./megam_opt -quiet -fvals multiclass train > f50.mem (for Cygwin)

2. **Test model *C***. Test the resulting model with classifier.py to compute the probability of each class for each input example, and produce the same output than megam test mode. Use the correct answer in the test files to compute the accuracy statistics.

   python ./classifier.py f50.mem <f50/test >out

3. Modify classifier.py to produce the probability of each class for each input example as output. Name the new version classifier-probs.py

4. Implement a program in python for **active learning** a Maximum Entropy Model.

   1. Use megam as base learned.
   2. Use the encoded corpus f50/train.f0 as initial training set D_L (keep these examples)
   3. Learn the initial model with D_L.
   4. Use corpus f50/hypothetically_unlabeled as it was the unlabeled data set D_U. (keep all these examples)
   5. Use classifier-probs.py and margin sampling to select the best K unlabeled examples (decide K).
   6. Use the annotated labels from f50/hypothetically_unlabeled to simulate the human annotation. Delete the selected example from D_U and add it to D_L.
   7. Learn a new model f_i.mem (0<i) using D_L.
   8. Use the classifier.py to compute the accuracy of model f_i.mem. Save the output results.

9. Stop when the new model performs as passively learning (i.e., model f50.mem).

5. Modify the previous program to perform entropy sampling instead of margin sampling .

6. How many examples are necessary with active learning to achieve the same accuracy than passive learning? To justify the answer, generate a figure showing learning curves for active learning with margin sampling and entropy sampling against the passive learning result. Take the output results achieved at each iteration in the active learning procedure to generate it.